CO542: Neural Networks and Fuzzy Systems Project Proposal

January 30, 2025

Title : Comment Toxicity Detection Model

Date : 30/01/2025

Team Members

- Dilshan D.M.T. (E/20/069)
- Rathnaweera R.V.C. (E/20/328)
- Karunarathne K.N.P. (E/20/189)
- Dilshan W.M.N. (E/20/455)

1 Problem Definition

1.1 Introduction

The rise of online platforms has enabled widespread communication, but it has also led to an increase in toxic comments, cyberbullying, and hate speech. These harmful interactions create unsafe environments, discourage engagement, and impact users' mental well-being. To address this, an automated comment toxicity detection model using neural networks is proposed.

1.2 Scope

The proposed model will classify user-generated comments as toxic or non-toxic and categorize toxicity into subtypes such as hate speech, threats, and offensive language. The system will be designed for real-time moderation and integration into social media platforms, forums, and online communities.

1.3 Justification for Using Neural Networks

Traditional rule-based and keyword-matching approaches struggle to detect nuanced toxicity, such as sarcasm and implicit hate speech. Neural networks, particularly deep

learning models like Convolutional Neural Networks (CNNs), Recurrent Neural Networks (RNNs), and Transformer-based architectures (e.g., BERT, RoBERTa), have shown superior performance in NLP tasks.
Transformer-based models like BERT and RoBERTa leverage self-attention mechanisms, enabling them to analyze entire text sequences in parallel. This makes them ideal for detecting complex language patterns, outperforming CNNs (which detect only local features) and RNNs (which struggle with long-range dependencies). Given their success in sentiment analysis and toxicity detection, a Transformer-based approach is justified.

2 Literature Review

Several studies have explored automated toxicity detection:

- **Jigsaw's Toxic Comment Classification Challenge (2018):** This competition provided a dataset of toxic comments for training deep learning models. Winning solutions leveraged LSTMs and Transformer-based models. `https://paperswithcode.com/paper/from-hero-to-z-eroe-a-benchmark-of-low-level`
- **BERT for Toxic Comment Detection (2020):** Research showed that pre-trained Transformer models outperform traditional approaches in detecting toxic language. `https://github.com/Mohammad8921/ToxicCommentDetection-FineTuningBert`
- **Multi-label Classification Approaches (2021):** Researchers have developed models capable of categorizing different types of toxicity simultaneously, improving moderation efficiency. `https://www.researchgate.net/publication/361051294_A_Systematic_Literature_Review_on_Multi-Label_Classification_based_on_Machine_Learning_Algorithms`
- **Explainable AI in Toxicity Detection (2023):** Recent studies have focused on improving interpretability, ensuring that AI-based decisions are transparent and fair.

3 High-Level Neural Network Model Design

3.1 Model Architecture

The proposed model leverages a fine-tuned Transformer-based model such as BERT or RoBERTa:

- **Input Layer:** Tokenized text representation using WordPiece tokenization.
- **Pre-trained Transformer Encoder:** Capturing contextual relationships between words.
- **Fully Connected Layers:** Classification into toxic and non-toxic categories, with potential for multi-label classification.
- **Softmax / Sigmoid Activation:** Output probability scores for toxicity levels.

3.2 Data Preprocessing

- Text Cleaning: Removing special characters, links, and emojis.
- Tokenization: Converting text into numerical sequences.
- Embedding Representation: Using pre-trained word embeddings.

3.3 Training and Evaluation

- **Dataset:** Jigsaw toxic comment dataset.
- **Loss Function:** Binary Cross-Entropy for toxicity detection, Multi-label loss for subtype classification.
- **Performance Metrics:** Accuracy, Precision, Recall, F1-score.
- **Regularization:** Dropout layers to prevent overfitting.

3.4 Model Architecture

4 Deployment : MLOps Integration Strategy

To ensure efficient deployment and lifecycle management, MLOps will be integrated using MLFlow:

- **Experiment Tracking:** Logging model versions and hyperparameters.
- **Model Versioning:** Managing different iterations of the model.
- **Automated Deployment:** Streamlining deployment to cloud or on-premise environments.
- **Monitoring**
  Maintenance: Tracking model performance to detect drift and retrain when necessary.

Project Timeline

Milestone 1: Domain Research and Proposal

Task	Details	Deadline
1. Problem Definition	Define the problem, scope, and justification for using neural networks.	15th January 2025
2. Literature Review	Research and summarize similar projects/research in the chosen domain.	20th January 2025
3. High-Level Model Design	Design the neural network architecture (e.g., Transformer-based models).	22nd January 2025

4. Proposal Draft	Compile the problem definition, literature review, and model design.	24th January 2025
5. Final Proposal Submission	Submit the detailed project proposal.	27th January 2025

Milestone 2: Implementation, Evaluation, and MLOps Integration

Task	Details	Deadline
1. Dataset Collection	Collect and preprocess the Jigsaw toxic comment dataset.	5th February 2025
2. Model Development	Implement the neural network model (e.g., BERT, RoBERTa).	15th February 2025
3. Model Training	Train the model using the dataset and fine-tune hyperparameters.	20th February 2025
4. Performance Evaluation	Evaluate the model using metrics (accuracy, precision, recall, F1-score).	25th February 2025
5. MLOps Integration	Integrate MLFlow for experiment tracking, model versioning, and deployment.	28th February 2025
6. Final Presentation Prep	Prepare slides and demo for the presentation.	1st March 2025
7. Peer Evaluation	Conduct peer evaluations and finalize the project.	3rd March 2025

Key Notes:

- **Weekly Team Meetings:** Schedule weekly meetings to track progress, discuss challenges, and assign tasks.
- **Task Allocation:** Assign tasks based on team members' strengths (e.g., coding, research, documentation).
- **Documentation:** Maintain a shared document (e.g., Google Docs) for progress tracking and collaboration.
- **MLOps Integration:** Ensure that MLFlow is set up early to log experiments and track model versions.

5 Conclusion

This project aims to build an effective comment toxicity detection model using deep learning techniques. By leveraging Transformer-based architectures, the model will enhance moderation on online platforms, reducing toxic content and fostering safer digital interactions.

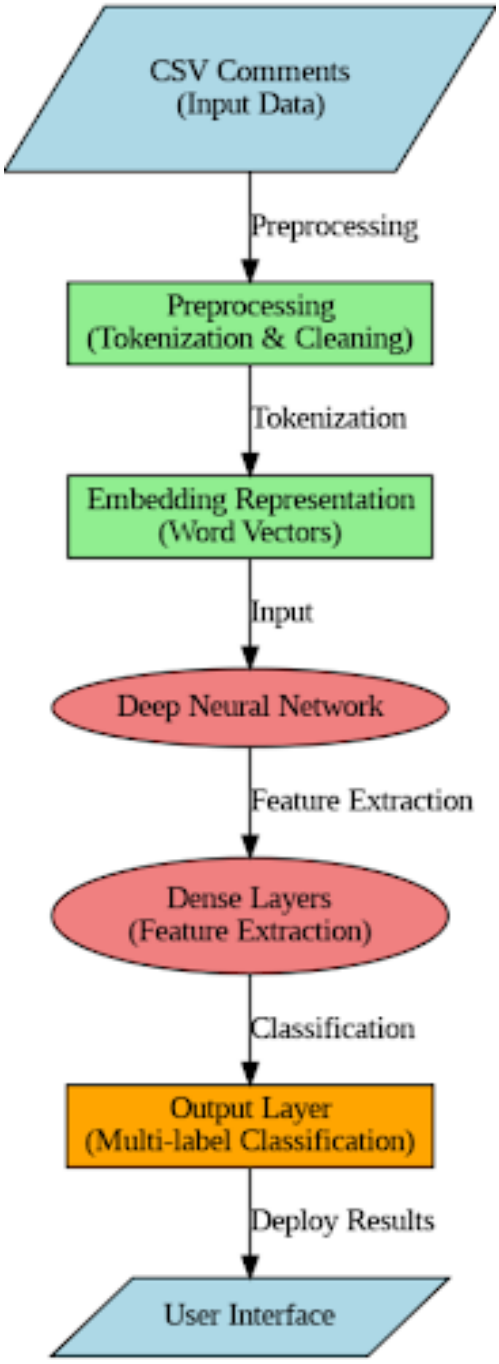

Figure 1: High-Level Model Architecture